UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,694	01/02/2004	Robert DeSantis	22233-08078	3817

24852 7590 12/04/2006

INTERNATIONAL BUSINESS MACHINES CORP
IP LAW
555 BAILEY AVENUE , J46/G4
SAN JOSE, CA 95141

EXAMINER	
NGUYEN, PHILLIP H	

ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 12/04/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/750,694	DESANTIS, ROBERT	
	Examiner	Art Unit	
	Phillip H. Nguyen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 January 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *02 January 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *20050314,20040402*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the original filling of January 2, 2004. Claims 1-20 are pending and have been considered below.

Claim Objections

2. Claim 3 is objected to because of the following informalities: Claim 3 recites, "creating a Java™ object having a same name as..." the examiner suggests the applicant to change the recited limitation to "creating a Java™ object having the same name as..."

3. Claims 1-20 are objected to because of the following informalities: Claims 1, 9, 10, 17, and 20 recite "API", but never defined "API" in the specification and the claims. The applicant is required to define "API" in all claims and specification. Also need to define JSP in claim 8 and SOAP in claims 14-16. For the examining purposes, the examiner interprets "API" as Application Program Interface, "JSP" as Java Server Page, and "SOAP" as Simple Object Access Protocol. Claims 2-7, 11-13 and 18-19 are directly or indirectly dependent of claims 1 and 17 respectively, and, therefore, have been addressed in connection with the objection of claims 1 and 17.

Appropriate correction is required.

Specification

4. The use of the trademark JavaScript™ and Java™ have been noted in this application. They should be capitalized wherever they appear and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner, which might adversely affect their validity as trademarks.

5. In paragraphs [0030] and [0053], applicant discloses "API" and "HTML" in paragraph [0021], [0023], [0025], [0046], [0053], and [0054] but never defined "API" and "HTML". The applicant is required to define "API" and "HTML". For examining purposes, the examiner interprets "API" as "Application Program Interface" and "HTML" as "HyperText Markup Language".

Appropriate correction is required.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Trademark or Trade Name as a Limitation in the Claims:

Claims 1-20 contain the trademark or trade name JavaScript™. Where a trademark or trade name is used in a claim as a limitation to identify or describe a

particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See Ex parte Simpson, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the good associated with the trademark or trade name. In the present case, the trademark or trade name is used to identify or describe a family of products generated in the propriety programming language call JavaScript™ and accordingly, the identification or description is indefinite.

8. Claim 5 recites the limitation "wherein the identifying include identifying ..." There is insufficient antecedent basis for this limitation in the claim. It is unclear to the Examiner whether identifying is new limitation of claim 5 or further limit its parent claim. However, examiner notices that "identifying" is not recited in claim 1. Claims 6-8 are directly depend on claim 5, and therefore, suffer the same deficiency as claim 5.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

10. Claims 1 and 4 are rejected under 35 U.S.C. 102(b) as being anticipated by Nelson (Patent No.: US 7,093,194 B2).

As per claim 1:

Nelson discloses a process for setting up the execution of the server-side method by a client-side data processing system, comprising:

a. although, Nelson does not explicitly disclose creating an API on the server side for a server-side software method. It is inherent in Nelson ("**API and TABLE I**" Col 5, line 31-65).

b. automatically creating a JavaScript™ program that contains calls to the server-side software method in accordance with the API ("**converter 27 generates the client-side script**" Col 8, line 53-54); and

c. sending the created JavaScript™ program to the client side ("**transmits the script to web browser via network**" Col 8, line 54-55).

As per claim 4:

Nelson discloses the process as in claim 1 above; and further discloses that wherein the JavaScript™ is executed by a non-modified standard browser program ("**Upon receiving the script, client device 9 executes the script and builds a presentation model 35" Col 8, line 55-57, this indicates that the client executes the script to build a presentation without modified itself**).

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 2, 12-13, and 17-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1), in view of Merrill et al. (Patent No.: US 6,369,821 B2).

As per claim 2:

Nelson discloses the process as in claim 1 above; and further discloses (**"upon receiving the script, client device 9 executes the script"** Col 8, line 56).

Nelson does not explicitly disclose executing the script on the client side to call the server side software method.

However, Merrill discloses an analogous process that executing the script on the client side to call the server side software method (**"to execute the script code, the browser used the interpreter to translate the code and then accesses the OLE control interface in response to references to the control interface in the script code...when the script code references the character control, the browser accesses the animation server"** Col 35, line 15, which means, when executing the

script on the client side, it invokes the server so that the browser can access the server).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to include the invoking the server process of Merrill in Nelson's approach. One of the ordinary skilled in the art would have been motivated to include the invoking the server process in Nelson because the browser can access the animation server's methods and properties of a particular character to the animation server (Col 35, line 15-16).

As per claim 12:

Nelson and Merrill disclose the process as in claim 2 above; and Merrill further discloses wherein executing the JavaScript™ includes converting the parameters sent to the server side ("**Clients invoke this method to instruct the server to generate speech output for a specified text string. Clients specify a text string, which the speech output engine converts into digitized audio output. The animation server plays clip**..." Col 23, line 1-5).

As per claim 13:

Nelson and Merrill disclose the process as in claim 2 above; and Merrill further discloses wherein executing the JavaScript™ includes converting results sent from the server side ("**The server generates this event when it encounters a bookmark tag**

in a text string as it converts the text string into speech output. The client can insert this tag in the text string provided with a Speak method" Col 24, line 58-61).

As per claim 17:

Nelson discloses a process for executing a server-side method by a client-side data processing system perform by the client-side data processing system and comprising:

a. receiving a JavaScript™ program that is automatically created by the server-side ("**converter 27 generates the client-side script, packet engine 21 transmits the script to web browser 33**" Col 8, line 53-55).

b. executing the JavaScript™ on the client side ("**upon receiving the script, client device 9 executes the script**" Col 8, line 55-56).

Nelson does not explicitly disclose JavaScript™ program...contains calls to the server-side software method in accordance with an API and executing the JavaScript™...to call the server-side software.

However, Merrill discloses an analogous process includes script contains calls to the server-side software method in accordance with an API that when executing the script on the client side, calling the server-side software methods ("**browser loads an interpreter in the process space of the browser. To execute the script code, the browser uses the interpreter to translate the code and then accesses the OLE control interface in response to references to the control interface in the script code... when the script code reference the character control, the browser**

accesses the animation server...access to the animation server's methods and properties for a particular character to the animation server" Col 35, line 1-15).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson's process to include calls to server-side's methods in the JavaScript™. One of the skilled in the art would have been motivated to modify Nelson's process because it allows browser to access animation server's methods and properties for a particular character to the animation server (Col 35, line 1-15).

As per claim 18:

Nelson and Merrill disclose the process as in claim 17 above; and Nelson further discloses passing parameters to the server side method when the JavaScript™ is executed on the client side ("**TABLE I" Col 5, line 35-65; the API (Application Program Interface) accepts all the parameters, for instance: vCubeID as integer, vReportID as string**...).

As per claim 19:

Nelson and Merrill discloses the process as in claim 17 above; and Nelson further discloses receiving results from the server-side method when the JavaScript™ is executed on the client side ("**upon receiving the script, client device 9 executes the script and builds a presentation model...In addition, client device 9 requests any remaining data that the OLAP servers returned...upon receiving the request,**

packet engine 21 transmits the remaining data in a series of packets to client device..." Col 8, line 56-60).

As per claim 20:

Nelson discloses an apparatus for setting up the execution of a server-side methods by a client-side data processing system, comprising computer instructions being executed by a processor, portions of the instructions comprising:

a. a portion for creating an API on the server side for a ser-side software method ("**API and TABLE I**" Col 5, line 31-65).

b. a portion for sending the created JavaScript™ program to the client side ("transmits the script to web browser" Col 8, line 54-55).

c. a portion for automatically creating a JavaScript™ program ("generates the client-side script" Col 8, line 54),

Nelson does not explicitly disclose JavaScript™ contains calls to the server-side software methods in accordance with the API.

However, Merrill discloses an analogous apparatus includes script that contains reference to server-side software method ("**browser loads an interpreter in the process space of the browser. To execute the script code, the browser uses the interpreter to translate the code and then accesses the OLE control interface in response to references to the control interface in the script code... when the**

script code reference the character control, the browser accesses the animation server...access to the animation server's methods and properties for a particular character to the animation server" Col 35, line 1-15).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson's process to include calls to server-side's methods in the JavaScript™. One of the skilled in the art would have been motivated to modify Nelson's process because it allows browser to access animation server's methods and properties for a particular character to the animation server (Col 35, line 1-15).

13. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1) and Merrill et al. (Patent No.: US 6,369,821 B2) as applied to claim 2 above, and further in view of Guthrie et al. (Patent No.: US 6,549,955 B2).

As per claim 3:

Nelson and Merrill disclose the process as in claim 2 above, but does not explicitly disclose wherein executing the JavaScript™ includes creating a Java™ object having the same name as a server-side Java™ bean.

However, Guthrie discloses an analogous process for creating a Java™ object having the same name as a server-side Java™ bean ("**remote proxy class (a**

category of objects) 23 is generated on client system based on the name, interfaces and methods of subject object 18 which my reside on server system...."reflection is a process that determines what an object can do, how it is defined... reflection mirrors the public view of an object to collect information...the reflection process includes the following: name, list of implemented interface; list of methods; and superclass information." Col 5, line 57-59; Col 6, line 39-56)

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson and Merrill's approaches to include the refection process. The modification is obvious because one of the ordinary skilled in the art would have been motivated to include the reflection process to **facilitate the creation of proxies, which resemble objects on the public view, but are very different internally, or privately** (Col 6, line 42-43).

14. Claims 5 and 7-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1), in view of Mikhail et al. (Pub No.: 2003/0218633 A1).

As per claim 5:

Nelson discloses the process as in claim 1 above, but does not explicitly disclose initially registering the server-side method on the server side, wherein the identifying includes identifying the registered server-side methods.

However, Mikhail discloses an analogous process for registering the server-side method on the server side, wherein the identifying includes identifying the registered server side methods ("**the bean registers itself with the Sybase notification server, specifying a callback method for the desired notification**" paragraph 0042).

Therefore, it would have obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson's approach to include registering beans on the server. The modification is obvious because one of the ordinary skilled in the art would have been motivated to register the beans on server for the server to remote the beans interface to the client in form of JavaScript™ object. This allows the programmer to call any of the bean's methods from the JavaScript™.

As per claim 7:

Nelson and Mikhail disclose the process as in claim 5 above; and Mikhail further discloses wherein a subset of methods are specified, thereby identifying the subset of methods ("**for each Sybase notification to be handled, the application server creates at least one Java™ bean. The bean registers itself...specifying a callback method for the desired notification**" paragraph 0042, which means, there are multiple beans are created and each specifying a callback method. The subset of methods is the subset of callback methods for these beans).

As per claim 8:

Nelson and Mikhail disclose the process as in claim 5 above, but does not explicitly disclose wherein registration is performed using JSP tags. It is inherent in Mikhail's process because JSP tags are included in Mikhail (paragraph 0034).

15. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1) and Mikhail et al. (Pub No.: 2003/0218633 A1) as applied to claim 5 above, and further in view of Chan et al. (Patent No.: US 6,836,889 B1).

As per claim 6:

Nelson and Mikhail discloses the process as in claim 5 above, but does not explicitly disclose wherein no methods are specified, thereby identifying all methods of a bean.

However, Chan discloses an analogous process includes no methods are specified, thereby identifying all methods of a beans ("**business methods of an enterprise bean that can be accessed by a client program**" Col 2, line 26-27).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to include this feature in Nelson and Mikhail's processes. The combination is obvious because one of the skilled in the art would have been motivated to include this feature to allow the tools and programmers to easily

recognize and utilize the capabilities of Java™ beans defined according to the Java™ Beans specification (Col 2, line 30-36).

16. Claims 9-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1), in view of Mandal et al. (Patent No.: US 7,043,732 B2).

As per claim 9:

Nelson discloses the process as in claim 1 above, but does not explicitly disclose the API (Application Program Interface) identifying parameters of the method.

However, Mandal discloses an analogous process of using the API (application Program Interface) identifying parameters of the method ("**provides APIs to add or remove a SNDRSet from the SNDRGroup, to lock or unclock a group, to modify group adjustable parameters**..." Col 16, line 42-44).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson's process to have his API (Application Program Interface) performs identify parameters. One of the skilled in the art would have been motivated to use API (Application Program Interface) for identifying parameters because API (Application Program Interface) allows programmer to add, remove, or modify parameters (Col 16, line 42-44).

As per claim 10:

Nelson discloses the process as in claim 1 above, but does not explicitly disclose allowing specification of which server-side methods are included in the API.

However, Mandal discloses an analogous process of using the API (Application Program Interface) and allowing specification of which server-side methods are included in the API ("**provides APIs to add or remove a SNDRSet (method) from the SNDRGroup (methods)...**" **Col 16, line 42-43, API (Application Program Interface) identifies which method to add or remove**).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson's process having API (Application Program Interface) to chose which method to include in API. One of the skilled in the art would have been motivated to modify because API (Application Program Interface) allows to remove or to add methods (Col 16, line 42-43).

As per claim 11:

Nelson discloses the process as in claim 1 above, but does not explicitly disclose the method is a method in a Java™ bean.

However, Mandal discloses an analogous process using method in a Java™ bean ("**the SNDR Federated bean 434 (SNDRBean) comprises an implementation 500 that is created by a constructor for** ..." Col 16, line 11-12, SNDRBean is a method that comprises an implementation).

Therefore, it would have been obvious to one having an ordinary skilled in the art at the time the invention was made to modify Nelson's approach to include a method in a Java™ bean. One of the ordinary skilled in the art would have been motivated to use a method of Java™ bean because it is a reusable software component and it can be combined to create an application.

17. Claims 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1) and Merrill et al. (Patent No.: US 6,369,821 B2) as applied to claim 2 above, and further in view of Mein et al (Patent No.: US 6,457,066 B1).

As per claim 14:

Nelson and Merrill disclose the process as in claim 2 above, but do not explicitly disclose wherein executing the JavaScript™ further includes using SOAP calls to invoke the server-side method.

However, Mein discloses an analogous process that using SOAP calls to invoke the server-side method ("**The protocol, called a Simple Object Access Protocol (SOAP), is an application layer protocol that is layered on top of HTTP and allows Microsoft Component Object Model (COM) Automation objects to be accessed and methods to be invoked over the internet through Web servers protected by firewalls**" Col 3, line 10-15").

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to include SOAP in Nelson and Merrill's approaches. One of the skilled in the art would have been motivated to include SOAP in Nelson and Merrill's approaches because using SOAP to access objects and to invoke methods (Col 3, line 10-15).

As per claim 15:

Nelson and Merrill disclose the process as in claim 2 above, but do not explicitly disclose wherein executing the JavaScript™ further includes using SOAP protocol to return results from the server-side method.

However, Mein discloses an analogous process for using SOAP protocol to return results from the server-side method ("**The SOAP stub, which is running on the Web server, unpacks and parses the SOAP request, instantiates the COM Automation object...The SOAP stub also returns any [out] or [in, out], or returns parameters from the COM Automation object instance to the SOAP proxy**..." Col 3, line 54-60).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to include SOAP in Nelson and Merrill's approaches. One of the skilled in the art would have been motivated to include SOAP in Nelson and Merrill's approaches because using SOAP to access objects and to invoke methods (Col 3, line 10-15).

As per claim 16:

Nelson discloses the process as in claim 1 above; and further discloses wherein the JavaScript™ includes JavaScript™ instructions for:

a. although, Nelson does not explicitly disclose declaring public methods for a current session. It is inherent in Nelson' process of generating the JavaScript™ in order for the browser to build the presentation model object ("**builds a presentation model**" Col 8, line 57).

instantiating JavaScript™ objects on the client-side that correspond to server-side objects ("**upon receiving the script, client device 9 executes the script and builds a presentation model**" Col 8, line 56-57, because the server-side device generated the script and the presentation model object builds based on the script, therefore, it is corresponding to server-side).

Nelson does not explicitly disclose setting up SOAP requests for each server-side method

However, Mein discloses an analogous process for setting up SOAP requests for each server-side method ("**The protocol, called a Simple Object Access Protocol (SOAP), is an application layer protocol that is layered on top of HTTP and allows Microsoft Component Object Model (COM) Automation objects to be accessed and methods to be invoked over the Internet through Web servers protected by firewalls**" Col 3, line 10-15).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to include SOAP in Nelson and Merrill's approaches.

One of the skilled in the art would have been motivated to include SOAP in Nelson and Merrill's approaches because using SOAP to access objects and to invoke methods (Col 3, line 10-15).

Conclusion

18. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. House et al (Patent No.: US 6,188,400 B1) discloses remote scripting of local objects.

b. Dillingham (Patent No.: US 6,327,608 B1) discloses server administrator tool using remote file browser.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Friday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
11/21/06

Wei Zhen
Supervisory Patent Examiner

Mary Steelman
Primary Examiner 11.27.2006